**WHAT IS CLAIMED IS:**

1. A system, comprising:

a master data server, to maintain a master database storing master data objects, the master data server using master identifiers to identify the master data objects, the master database being accessible to clients; and

an integration server, in response to a request from a client to access master data identified by a client identifier, to map the client identifier to a master identifier, retrieve a master data object from the master database based on the master identifier, and map the master data object to a mapped data object based on a set of mapping rules associated with the client.

2. The system of claim 1, wherein at least two clients use different client identifiers to identify a common master data object.

3. The system of claim 1, further comprising a mapping table to store information related to the mapping of the client identifiers to the master identifiers.

4. The system of claim 1, further comprising a mapping table to store mapping rules associated with the clients.

5. The system of claim 1, wherein the master data object has a plurality of attributes associated with characteristics of an entity represented by the master data object, and mapping the master data object to the mapped data object comprises retrieving a subset of the attributes from the master data object and formatting the subset of attributes based on rules defined by the client.

6. The system of claim 1, wherein the integration server dynamically maps the master data object in the master database to the mapped data object based on mapping rules defined by the client each time the client requests for the master data without replicating the master data object at a database local to the client.

7. The method of claim 1, wherein the integration server comprises a cache to store master data objects that are requested by clients, and to provide stored master data objects to clients when the integration server receives requests that are identical to previous requests for access to the master data objects.

8. The system of claim 1, wherein the integration server comprises an exchange interface receives data that are published by a first client, and routes the published data to a second client that requested the published data.

9. The system of claim 8, wherein the integration server maps the data published by the first client to master data based on a first set of mapping rules associated with the first client, and maps the master data to mapped data that can be processed by the second client based on a second set of mapping rules associated with the second client.

10. The system of claim 1, wherein the integration server comprises a content integrator that finds characteristics that at least two clients associate with an object.

11. The system of claim 1, wherein the integration server comprises an adapter that receives communications from a client and extracts master data from the communications and forwards the extracted master data to the master data server.

12. The system of claim 1, wherein the master data server sends master data objects requested by clients to the clients without performing client authorization checks.

13. The system of claim 12, wherein the client performs authorization checks to limit access to the master data objects by users.

14. The system of claim 1, wherein the master data server performs authorization checks to limit access to the master data objects by users.

15. The system of claim 12, wherein the client performs authorization checks to limit access to the master data objects by processes running on the client.

16. The system of claim 1, wherein the master data server provides processes to allow the clients to modify the master data.

17. The system of claim 1 in which a portion of the master data objects are associated with products.

18. The system of claim 1 in which a portion of the master data objects are associated with business partners.

19. A system, comprising:

a master data server, to maintain a master database storing master data objects, each object having a set of attributes, the master database being accessible to clients, each client processing a subset of attributes of the master data objects; and

an integration server, in response to a request from any one of the clients to access a master data object, to retrieve the master data object from the master database and map the master data object to a mapped data object based on a set of mapping rules associated with the client so that the mapped data object contains the subset of attributes in a format that can be processed by the client.

20. A method, comprising:

maintaining a master database at a data server, the master database containing master data objects, the master database accessible to clients;

receiving a request from a client to access master data, the request containing a client identifier;

mapping the client identifier to a master identifier;

retrieving a master data object based on the master identifier;

mapping the master data object to a mapped data object based on a set of mapping rules associated with the client; and

sending the mapped data object to the client.

21. The method of claim 20, further comprising receiving a request from the client to modify the master data object to create a modified master data object, and querying the other clients to verify that the modified master data object conforms to consistency rules defined by the other clients.

22. The method of claim 21, further comprising, if a particular client does not respond to the query as to whether the modified master data object conforms to consistency rules defined by the particular client, placing the particular client on an exception list to indicate that the modified master data object has not been verified to conform with the set of consistency rules defined by the particular client.

23. The method of claim 22, further comprising, after a predefined period of time or when the particular client attempts to access data in the database, performing another attempt to verify whether the modified master data object conforms to the consistency rules defined by the particular client.

24. The method of claim 20, further comprising receiving a request from the client to delete the master data object from the master database, querying the other clients to verify that the master data object is not used by the other clients, and deleting the master data object from the master database after confirming the master data object is not used by the other clients.

25. The method of claim 20, further comprising storing a master data object in a cache, and retrieving the master data object from the cache rather than from the master database when a request for access to the master data object is identical to a previous request.

26. The method of claim 20, further comprising modifying the master data objects in the master database, and modifying the mapping rules to allow the clients to process modified master data objects without making modifications at the client.

27. The method of claim 20, wherein each master data object has attributes, each client processes a subset of the attributes, different clients process different subsets of the attributes, and the mapping rules associated with a client define which subset of attributes are processed by the client.

28. A method for maintaining data comprising:

providing a master database having master data shared by at least two clients;

providing an interface for updating the master database;

providing an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client; and

providing a user interface for entering and displaying subsets of the master data.

29. The method of claim 28, further comprising providing an exchange infrastructure that receives published data, published by a client, and routes the published data to another client that has requested the published data.

30. The method of claim 29, further comprising providing a content integrator to find characteristics that a first client and a second client commonly associate with an object.

31. The method of claim 30, further comprising receiving updates of the characteristics for an object from either one of the first and second clients, and sending the updates to the other of the first and second clients.

32. The method of claim 28, further comprising providing a content integrator to find characteristics that at least two clients associate with an object.

33. The method of claim 28, further comprising dynamically mapping the data in the master database to mapped data having a format conforming to rules defined by the client each time the client requests for data without replicating data stored in the master database to a database local to the client.

34. The method of claim 28, further comprising receiving updates of the characteristics for an object from either one of the first and second clients, and sending the updates to the other of the first and second clients.

35. A method for maintaining data, comprising:

receiving a first identifier used by a first client to identify a data object, and a request to delete the data object, the data object being stored in a database maintained by a data server, the database being accessible to the first client and a second client;

mapping the first identifier to a second identifier used by the second client to identify the data object;

mapping the first identifier to a third identifier used by the data server to identify the data object;

querying the second client based on the second identifier to determine whether the second client is using the data object; and

if the second client is not using the data object, deleting the data object from the database based on the third identifier.

36. The method of claim 35 in which querying the second client includes determining whether there is any reference to the data object in processes running on the second client and whether there is any reference to the data object in data buffers of the second client.

37. The method of claim 35 further comprising querying the second client to determine whether the second client objects to deletion of the data object, and preventing deletion of the data object if the second client objects.

38. The method of claim 35 in which the data object has a plurality of attributes, the first client configured to access a first subset of the attributes, the second client configured to access a second subset of the attributes, the second subset being different from the first subset.

39. A method comprising:

receiving a first set of communications from a first client;

analyzing the first set of communications to find a set of characteristics that the first client associates with a data object used in the first set of communications;

analyzing other communications received from clients to find additional sets of characteristics that clients associate with data objects that have the same characteristics as the first set of characteristics;

placing the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group; and

generating a data distribution path to allow updates of the set of characteristics to be sent to the client group.

40. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

maintain a master database at a data server, the master database containing master data objects, the master database accessible to clients;

receive a request from a client to access master data, the request containing a client identifier;

map the client identifier to a master identifier;

retrieve a master data object based on the master identifier;

map the master data object to a mapped data object based on a set of mapping rules associated with the client; and

send the mapped data object to the client.

41. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

maintain a master database having master data shared by at least two clients;

provide an interface for updating the master database;

provide an interface for mapping subsets of the master data into mapped data having a format that is acceptable to each client; and

provide a user interface for entering and displaying subsets of the master data.

42. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

receive a first identifier used by a first client to identify a data object, and a request to delete the data object, the data object being stored in a database maintained by a data server, the database being accessible to the first client and a second client;

map the first identifier to a second identifier used by the second client to identify the data object;

map the first identifier to a third identifier used by the data server to identify the data object;

query the second client based on the second identifier to determine whether the second client is using the data object; and

if the second client is not using the data object, delete the data object from the database based on the third identifier.

43. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

receive a first set of communications from a first client;

analyze the first set of communications to find a set of characteristics that the first client associates with a data object used in the first set of communications;

analyze other communications received from clients to find additional sets of characteristics that clients associate with data objects that are the same characteristics as the first set of characteristics;

place the first client and clients who sent a set of characteristics that are the same as the first set of characteristics into a client group; and

generate a data distribution path so that the programmable processor can route updates of the set of characteristics to the client group.

44. A computer program product, tangibly stored on a machine-readable medium, for dynamic access of master data, comprising instructions operable to cause a programmable processor to:

associate master data with an object;

send the master data to a master data server that stores master data associated with the object on a database; and

access master data associated with objects on the database by requesting that an integration server that communicates with the programmable processor and the master data server map the data in the data server to a mapped data set that has a format conforming to rules defined by the programmable processor and send the mapped data set to the programmable processor.

45. The computer program product of claim 44 wherein the integration server communicates with the programmable processor and the master data server dynamically.

46. The computer program product of claim 44 wherein the programmable processor sends a set of data to an exchange infrastructure that sends the set of data to another programmable processor that has requested the set of data.

47. The computer program product of claim 44 wherein the programmable processor sends characteristics that it associates with a data object to a content integrator which finds other programmable processors that associate the characteristics with the data object.

48. The computer program product of claim 44 wherein the programmable processor sends communications to an adapter that extracts master data from the communications and forwards the extracted master data to the master data server.

49. The computer program product of claim 44 wherein the master data server sends all data requested by the programmable processor to the programmable processor without performing programmable processor authorization checks.

50. The computer program product of claim 44 wherein the programmable processor can modify the master data stored in the master data server.